WHAT IS CLAIMED IS:

1   1.   A method for recoverable programming, comprising the steps of:

2   identifying a predetermined instruction sequence;

3   monitoring for memory access errors;

4   logging a memory access error in an error logging register;

5   polling the register for any logged memory access error during execution of the

6   instruction sequence; and

7   raising exceptions, if the memory access error is logged.

1   2.   The method of claim 1, further comprising the steps of:

2   checkpointing a predetermined set of system data; and

3   recovering from the memory access error using the checkpointed system data, if

4   the memory access error is logged during execution of the instruction sequence.

1   3.   The method of claim 1, further comprising the step of:

2   setting data returned in response to the memory access request equal to a set of

3   predefined fake data, if the memory access error is logged during execution of the

4   instruction sequence.

1   4.   The method of claim 3, further comprising the step of:

2   skipping the polling and raising steps if the data returned in response to the

3   memory access request is not equivalent to the predefined fake data.

1   5.   The method of claim 1, further comprising the step of:

15

2   masking a machine check abort handle.

1   6.   The method of claim 5, after the raising step, further comprising the steps of:

2   enabling the machine check abort handle.

1   7.   The method of claim 1, further comprising the step of:

2   updating pointers, if the memory access error is logged.

1   8.   The method of claim 1, further comprising the step of:

2   re-executing the memory access request, if software so commands.

1   9.   A method for recoverable programming, comprising the steps of:

2   identifying a predetermined instruction sequence;

3   checkpointing a predetermined set of system data;

4   masking a machine check abort handle;

5   monitoring for memory access errors;

6   logging a memory access error in an error logging register;

7   polling the register for any logged memory access error during execution of the

8   instruction sequence;

9   raising exceptions, if the memory access error is logged;

10   updating pointers, if the memory access error is logged;

11   recovering from the memory access error using the checkpointed system data, if

12   the memory access error is logged during execution of the instruction sequence.;

13   re-executing the memory access request, if software so commands; and

16

14   enabling the machine check abort handle.

1   10.   A computer-usable medium embodying computer program code for commanding

2   a computer to perform recoverable programming, comprising the steps of:

3   identifying a predetermined instruction sequence;

4   monitoring for memory access errors;

5   logging a memory access error in an error logging register;

6   polling the register for any logged memory access error during execution of the

7   instruction sequence; and

8   raising exceptions, if the memory access error is logged.

1   11.   The medium of claim 10, further comprising the steps of:

2   checkpointing a predetermined set of system data; and

3   recovering from the memory access error using the checkpointed system data, if

4   the memory access error is logged during execution of the instruction sequence..

1   12.   The medium of claim 10, further comprising the step of:

2   setting data returned in response to the memory access request equal to a set of

3   predefined fake data, if the memory access error is logged during execution of the

4   instruction sequence.

1   13.   The medium of claim 13, further comprising the step of:

2   skipping the polling and raising steps if the data returned in response to the

3   memory access request is not equivalent to the predefined fake data.

1 14. The medium of claim 10, further comprising the step of:

2 masking a machine check abort handle.

1 15. A system for recoverable programming, comprising:

2 means for identifying a predetermined instruction sequence;

3 means for monitoring for memory access errors;

4 means for logging a memory access error in an error logging register;

5 means for polling the register for any logged memory access error during

6 execution of the instruction sequence; and

7 means for raising exceptions, if the memory access error is logged.

1 16. The system of claim 15, further comprising:

2 means for checkpointing a predetermined set of system data; and

3 means for recovering from the memory access error using the checkpointed

4 system data, if the memory access error is logged during execution of the instruction

5 sequence..

1 17. The system of claim 15, further comprising:

2 means for setting data returned in response to the memory access request equal to

3 a set of predefined fake data, if the memory access error is logged during execution of the

4 instruction sequence.

1 18. The system of claim 17, further comprising:

2 means for bypassing the means for polling and means for raising if the data

3 returned in response to the memory access request is not equivalent to the predefined fake

4 data.

1 19. The system of claim 15, further comprising the step of:

2 means for masking a machine check abort handle.